

1       **In the Claims**

2       Claims 21-36, 47-61, 64-70 and 82-96 are canceled without prejudice.

3       Claims 62 and 63 were previously canceled.

4       Claims 1-20, 37-46 and 71-81 remain in the application and are listed  
5 below as follows:

6  
7       1.     (Previously Presented) A method for delivering software via a  
8 network comprising:

9       describing one or more software extensions using a hierarchical language,  
10 the extensions being configured for incorporation on a client, said describing  
11 defining one or more manifests containing at least one list of files comprising an  
12 extension; and

13       delivering the one or more manifests to the client via the network, the one  
14 or more manifests being configured for use in downloading the software  
15 extensions via the network, at least some of the extensions being downloadable by  
16 streaming extension files to the client in a manner that enables a user to begin to  
17 interact with the extension sooner than if the user had to wait for the entire  
18 extension to load, said manner being developed based on scenario runs in which  
19 files that are more likely to be first used by the user are downloaded before files  
20 that are less likely to be first used, and wherein files that are less likely to be used  
21 first can be downloaded via a background download process.

22  
23       2.     (Original) The method of claim 1, wherein the one or more  
24 manifests are configured to assist in organizing delivery of individual files listed in  
25 the one or more manifests.

1  
2 3. (Original) The method of claim 1, wherein the one or more  
3 manifests are configured to assist in validating individual files listed in the one or  
4 more manifests.

5  
6 4. (Original) The method of claim 1, wherein the one or more  
7 manifests are configured to assist in updating individual files listed in the one or  
8 more manifests.

9  
10 5. (Original) The method of claim 1, wherein the one or more  
11 manifests describe individual file locations.

12  
13 6. (Original) The method of claim 1, wherein the one or more  
14 manifests contain individual hashes for one or more of the listed files.

15  
16 7. (Original) The method of claim 1, wherein the one or more  
17 manifests contain download directives for downloading the listed files.

18  
19 8. (Original) The method of claim 1, wherein the one or more  
20 manifests are defined in a tag-based language.

21  
22 9. (Original) The method of claim 1, wherein the one or more  
23 manifests are defined in extensible markup language (XML).

1           10.   (Original) The method of claim 1, wherein the network comprises  
2 the Internet.

3  
4           11.   (Original) One or more computer-readable media comprising  
5 computer-readable instructions thereon which, when executed by a computer,  
6 cause the computer to implement the method of claim 1.

7  
8           12.   (Original) A computer programmed with instructions which, when  
9 executed by the computer, implemented the method of claim 1.

10  
11          13.   (Previously Presented) One or more computer-readable media  
12 comprising computer-readable instructions thereon which, when executed by a  
13 computer, cause the computer to:

14           describe one or more software extensions using extensible markup  
15 language (XML), the extensions being configured for incorporation on a client,  
16 said describing defining a manifest containing at least one list of files comprising  
17 an extension, the manifest being configured to assist in one or more of the  
18 following: organizing delivery of individual files listed in the manifest, validating  
19 individual files listed in the manifest, and updating individual files listed in the  
20 manifest; and

21           deliver the manifest to the client via the network, at least some of the  
22 extensions being downloadable by streaming extension files to the client in a  
23 manner that enables a user to begin to interact with the extension sooner than if the  
24 user had to wait for the entire extension to load, said manner being developed  
25 based on scenario runs in which files that are more likely to be first used by the

1 user are downloaded before files that are less likely to be first used, and wherein  
2 files that are less likely to be used first can be downloaded via a background  
3 download process.

4  
5 14. (Previously Presented) A method for receiving software via a  
6 network comprising:

7 receiving a manifest that contains at least one list of files comprising a  
8 software extension that is to be downloaded via a network and incorporated on a  
9 client, the manifest being defined in extensible markup language (XML), the  
10 manifest being configured to assist in:

11 organizing delivery of the files,

12 validating individual files listed in the manifest, and

13 updating individual files listed in the manifest; and

14 downloading files from the list of files contained in the manifest;

15 wherein the extension is downloadable by streaming extension files to the  
16 client in a manner that enables a user to begin to interact with the extension sooner  
17 than if the user had to wait for the entire extension to load, said manner being  
18 developed based on scenario runs in which files that are more likely to be first  
19 used by the user are downloaded before files that are less likely to be first used,  
20 and wherein files that are less likely to be used first can be downloaded via a  
21 background download process.

22  
23 15. (Original) The method of claim 14, wherein the software extension  
24 is to be incorporated into a software platform executing on the client.  
25

1           16.    (Original) The method of claim 14, wherein the downloading of the  
2 files takes place by downloading the files in an order that is described in the  
3 manifest.

4  
5           17.    (Original) The method of claim 14 further comprising validating one  
6 or more downloaded files using the manifest.

7  
8           18.    (Original) The method of claim 14 further comprising updating one  
9 or more files using the manifest.

10  
11           19.    (Original) One or more computer-readable media comprising  
12 computer-readable instructions thereon which, when executed by a computer,  
13 cause the computer to implement the method of claim 14.

14  
15           20.    (Original) A computer programmed with instructions which, when  
16 executed by the computer, implement the method of claim 14.

17  
18           21.-36. (Canceled)

19  
20           37.    (Previously Presented) A method of providing software via a  
21 network comprising:

22           describing one or more software extensions using one or more extensible  
23 markup language (XML) files, the extensions being configured for incorporation  
24 in a software program executing on a client, individual XML files providing  
25 individual manifests that contain a list of files that comprise an extension; and

1 storing the XML files in a Web-accessible location;

2 wherein at least some of the extensions are downloadable by streaming  
3 extension files to the client in a manner that enables a user to begin to interact with  
4 the extension sooner than if the user had to wait for the entire extension to load,  
5 said manner being developed based on scenario runs in which files that are more  
6 likely to be first used by the user are downloaded before files that are less likely to  
7 be first used, and wherein files that are less likely to be used first can be  
8 downloaded via a background download process.

9  
10 38. (Original) The method of claim 37 further comprising storing  
11 extension files associated with the XML files in a Web-accessible location.

12  
13 39. (Original) The method of claim 37 further comprising:  
14 storing extension files associated with the XML files in a Web-accessible  
15 location; and  
16 providing one or more XML files and one or more associated extension  
17 files to a client via the network.

18  
19 40. (Original) The method of claim 37, wherein individual manifests can  
20 contain one or more file hashes that can be used for file security.

21  
22 41. (Original) The method of claim 37, wherein individual manifests can  
23 contain one or more file hashes that can be used for versioning.

1           42.   (Original) The method of claim 37, wherein individual manifests  
2 comprise one or more file groups that determine when particular files are  
3 downloaded to the client.

4  
5           43.   (Original) The method of claim 42, wherein the file groups  
6 determine where files are stored on the client.

7  
8           44.   (Original) The method of claim 42, wherein the file groups  
9 determine how files are packaged.

10  
11          45.   (Original) The method of claim 42, wherein the file groups  
12 determine where files are stored on the client and how files are packaged.

13  
14          46.   (Original) One or more computer-readable media having computer-  
15 readable instructions thereon which, when executed by a computer, cause the  
16 computer to implement the method of claim 37.

17  
18          47.-70. (Canceled)

19  
20          71.   (Previously Presented) An automated software tool comprising a  
21 package manifest creation tool configured to:

22               receive one or more input parameters pertaining to a package manifest that  
23 is to describe a software extension that is configured to extend a software  
24 application executing on a client; and  
25

1 generate a package manifest that describes the extension, the package  
2 manifest being generated using a hierarchical language;

3 wherein the extension is downloadable by streaming extension files to the  
4 client in a manner that enables a user to begin to interact with the extension sooner  
5 than if the user had to wait for the entire extension to load, said manner being  
6 developed based on scenario runs in which files that are more likely to be first  
7 used by the user are downloaded before files that are less likely to be first used,  
8 and wherein files that are less likely to be used first can be downloaded via a  
9 background download process.

10  
11 72. (Original) The automated software tool of claim 71, wherein the  
12 hierarchical language comprises a tag-based language.

13  
14 73. (Original) The automated software tool of claim 71, wherein the  
15 hierarchical language comprises extensible markup language (XML).

16  
17 74. (Original) The automated software tool of claim 71, wherein one  
18 input parameter specifies a directory containing files that are to be described in the  
19 package manifest.

20  
21 75. (Original) The automated software tool of claim 71, wherein one  
22 input parameter comprises file group information and load dependencies.

23  
24 76. (Original) The automated software tool of claim 71, wherein one  
25 input parameter comprises file usage statistics.



1  
2 77. (Original) The automated software tool of claim 76, wherein the file  
3 usage statistics are ascertained from scenario runs.  
4

5 78. (Original) The automated software tool of claim 77, wherein  
6 individual scenarios have individual priorities.  
7

8 79. (Original) The automated software tool of claim 76, wherein the file  
9 usage statistics are ascertained from scenario runs that are collected by running  
10 logs on various scenarios.  
11

12 80. (Original) The automated software tool of claim 76, wherein the file  
13 usage statistics are ascertained from scenario runs that are collected by running  
14 logs on various scenarios, the scenarios having individual checkpoints that  
15 separate one scenario from another.  
16

17 81. (Previously Presented) The automated software tool of claim 76,  
18 wherein the file usage statistics are ascertained dynamically by building a  
19 knowledge base that describes tasks that users typically accomplish.  
20

21 82.-96 (Canceled)  
22  
23  
24  
25